

ChiliProject - Feature # 139: Tags for issues

Status:	Open	Priority:	Normal
Author:	Aleksey Zapparov	Category:	Issue tracking
Created:	2011-02-04	Assignee:	
Updated:	2013-11-06	Due date:	
Remote issue URL:			
Affected version:			
Description:	Redmine missed this feature so much, so, there are at least 5 plugins for Redmine implementing this feature (including mine). And all they suck. So if there will be native support for tags - it would be awesome!		
	I have a fork of redmine with such feature in it (needs only few updates to migrate from act_as_taggable to act_as_taggable_on_steroids), so I can propose patches with tests if needed.		

Associated revisions

2011-12-10 11:31 am - Felix SchÄrfer

Merge pull request #139 from esmarkowski/patch-1

[#761] Fix quotation of since date in git adapter

History

2011-02-05 01:11 am - Eric Davis

Even I have a plugin for it: https://github.com/edavis10/redmine_tags. Mine was working on MySQL but the tagging library had a bug on my production server's Postgres so I can't run it in production.

2011-02-06 09:40 pm - Aleksey Zapparov

Yes, mine one was inspired by your one. :) Even named the same: https://github.com/iixti/redmine_tags :))

Now, I have finished all the things I wanted for myself. And plugin works fine with SQLite3 and PostgreSQL. I guess the problem you have experienced with AATO was fixed.

2011-02-06 10:44 pm - Aleksey Zapparov

Just tested with MySQL. I have some mistake in SQL query so will fix it ASAP. So strange this SQL works with SQLite3 and PostgreSQL. But don't work with MySQL. %|

2011-02-07 10:16 am - Stefan HÄrllÄn

Keep in mind that there are currently several different "tagging" plugins for Redmine, and many of them use tags for different things, but have database collisions. Which is bad, mkey? :)

Tagging of issues, wiki pages and contacts (in contacs plugin) all makes very much sense to how I use Redmine, and I think tagging basically should move into core so that plugins can use the then existing tagging system in any way they want, without colliding with each other.

2011-02-07 02:37 pm - Aleksey Zapparov

There's a ruby gem lies under the tagging plugins made by Eric Davis (as well as I also use the same gem). It called ActsAsTaggableOn. It give you ability to tag items without colliding. So Wiki.tags will never mess with Issue.tags, as they belongs to different taggable entities. :))

2011-02-08 12:22 am - Aleksey Zapparov

OK. Now my version of +redmine_tags+ works fine on SQLite3, PostgreSQL and MySQL. The only things it still miss are: filter columns and probably filter statement generation of raw SQL instead of relying on @Issue.tagged_with@. So now I can start on Chiliproject with tags branch :))

2011-02-08 01:28 am - Eric Davis

Stefan HÄrllÄn wrote:

> Tagging of issues, wiki pages and contacts (in contacs plugin) all makes very much sense to how I use Redmine, and I think tagging basically should move into core so that plugins can use the then existing tagging system in any way they want, without colliding with each other.

Yea that is what I'm thinking. Using the library in my plugin I was also able to add tags to users all within the same database schema. If I remember correctly, I just needed to declare "allow tags on users" and then hook up the user interface. It also also allowed tag groups so you could do something like:

- * issues have basic tagging
- * issues also have tags for "review status"
- * issues also have tags for "skills required"

2011-02-08 08:13 pm - Aleksey Zapparov

Yes. ActsAsTaggableOn RoR plugin allows to have more than one group of tags for each model. And tags are not clashed. Tags (in this plugin) have it's type, which is the model name, e.g. @Issue@, and context, which is by default @tags@. So you can have more than one context per model, and tags from different contexts as well as those belonging to different entities will never reach each other :))

2011-02-08 10:02 pm - Eric Davis

I think a good way to implement this is to:

- # add the tagging libraries to the core
- # create separate plugins to add tagging for the models. This way we can make sure the tagging in ChiliProject is modular and the plugin API works. I think 3 plugins would be good to ship with the core:
- #* issue tagging
- #* wiki page tagging
- #* forum thread tagging

Thoughts?

2011-02-08 10:14 pm - Aleksey Zapparov

I think that's a reasonable point of view. But as you know there are few very tight places in the core. For example there is no simple way to add your special filtering logic. So for example in my redmine_tags I had to use some kind of hack - https://github.com/ixti/redmine_tags/blob/master/lib/redmine_tags/patches/query_patch.rb - take a look at the statement_extended. There I had to first remove filter from the stack, then grab default filtering query, then put filter back to the stack and process it...

PS don't look at the TODO comment. After some time of thinking I decided that I'll leave this part as it is, as I don't see any point for too-damn-complex and too-deep-sub-selects-with-lots-of-joins.

UPDATE: Also, I think if we are going to have this features as separate core plugins, then we need to add some more -hacks- hooks for example, because I believe that tags should come right after issue description.

2011-02-08 10:42 pm - Eric Davis

Aleksey Zapparov wrote:

> I think that's a reasonable point of view. But as you know there are few very tight places in the core. For example there is no simple way to add your special filtering logic.

Yea, I've done similar things. I have enough hacks for @Query@ in my plugins that I think can start going through it and adding/refactoring it's public API a bit.

> UPDATE: Also, I think if we are going to have this features as separate core plugins, then we need to add some more hacks for example, because I believe that tags should come right after issue description.

We can add some plugin hooks for things as needed.

2011-02-09 10:33 am - Aleksey Zapparov

Eric Davis wrote:

> We can add some plugin hooks for things as needed.

Yes. I think we have to :) Sorry misused the word, I meant "more hooks" instead of "more hacks" indeed.

Just thought that we don't need three core plugins. I think it should be one core plugins, e.g. @chili_tags@, which will have tagging support for issues, wikis and forum threads. But to give flexibility, administrator will be able to disable tag support for any of them.

I think it's the best way to do so, as in fact if we will split it into three separate plugins we will repeat ourselves three times. Or will have to have separate fourth plugin to have common logic in it.

2011-04-12 09:56 am - Pedro Gutierrez

Very interesting plugin, thanks to everybody involved in these series of plugins

I'm happy to see that future version 1.3 will consider project specific tags: great for installations with a significant number of projects

I miss, however, one functionality I've seen in other similar plugins such as https://github.com/friflaj/redmine_tagging

I mean, the possibility of including tags inline in the issue description text in the following way

```
<pre>
{{tag (tag1, tag2)}}
</pre>
```

We find it helpful to aid automatic tagging of certain issues

2011-04-15 08:24 am - Pedro Gutierrez

- *File Issues_by_tag_within_a_version.jpg added*

For your consideration

Currently, see the picture below, it is not possible to filter the issues of a version by tag.

In my case, categories are not enough to classify my issues and so, being able to filter by tag would be a great enhancement

!Issues_by_tag_within_a_version.jpg!

2011-04-15 10:06 am - Aleksey Zapparov

Hello,

PG> I mean, the possibility of including tags inline in the issue description text in the following way

Can you explain, how this is easier then specifying tags inside special field? Or did you mean that it makes it possible to do it like this, e.g. description:

So we tried new {{way}} of {{thinking}}, in order to achieve better {{quality}}.

Will remove ``{` and `}`` and will put words wrapped by this delimiters to the tags?

I like your idea about showing info about issues per tag in version - I'll try to implement it.

2011-04-15 11:05 am - Pedro Gutierrez

Thanks Aleksey, for considering the possibility of showing info about issues per tag in version.

Aleksey Zapparov wrote:

> PG> I mean, the possibility of including tags inline in the issue description text in the following way

>

> Can you explain, how this is easier then specifying tags inside special field?

The special field for adding tags is perfect. It is simple and easy, no problem with it.

My suggestion is not to remove this special field (far from it) but to suggest a complementary method to add tags.

It might be that we are a special case, but a significant number of our issues descriptions is generated automatically and hence, the possibility of adding a tag just by including a string in the description is an easy way for our scripts to deal with tags.

2011-04-15 03:45 pm - Eric Davis

Once we have tags I want them to be a separate field. That way it's easy to query on them and access them programatically (e.g. via plugins). If we want to have a macro to add tags *in addition*, that could work by just parsing the body/comment for any new tags and adding them to the issue directly.

2011-04-15 05:56 pm - Aleksey Zapparov

Eric,

> If we want to have a macro to add tags in addition, that could work by just parsing the body/comment for any new tags and adding them to the issue directly.

I absolutely agree. Basically I meant that in my previous post. :))

2011-12-28 12:41 am - Eric Davis

The big blocking question I have for issue tagging is:

Should adding/removing tags on an issue be logged to the history?

The only reasons I can think of not logging tags to history are:

* then extra email notifications won't be sent

* then we can add/remove tags using Ajax and not have to refresh the page (much like how the related issues work currently)

2011-12-28 12:48 am - Aleksey Zapparov

I believe there's no need for putting tags on the journal. Tags are used for "additional" description, for "important" we can use categories IMHO.

2012-02-08 06:09 am - Tom Rochette

Eric Davis wrote:

> The big blocking question I have for issue tagging is:

>

> Should adding/removing tags on an issue be logged to the history?

>

> The only reasons I can think of not logging tags to history are:

>

> * then extra email notifications won't be sent

> * then we can add/remove tags using Ajax and not have to refresh the page (much like how the related issues work currently)

It's always possible to have the AJAX feature while still having the user confirm the tag changes. This way you don't have a messy change history.

This would work in the similar fashion we currently update or add comments, that is:

1. Click on update => Displays the comment section with tag somewhere
2. The user start typing words in the tag field, which get autocompleted
3. The user submit the form, which sends the tag field information once, allowing journaling to occur if it is decided it's wanted.

2012-02-08 01:41 pm - Victor Sergienko

I can imagine viable workflows that might need logged or unlogged tags, so it's just a matter of choice (unless one wants to clutter Chili with options).

Unlogged tags are ways better than none, plus there are implementation reasons to make them unlogged. I believe it basically resolves the blocking question :D

Tags could also be implemented in a way to make "tag history" plugin possible - one that enables tags history log.

2012-03-12 09:45 pm - Paweł, O

My 1 dollar concerning tagging.

The main question is what do we need tags for? Are they just for additional information? Are they for sorting? What are they for and who is going to be allowed for adding tags? If we answer these and similar questions than we will easily find the answer if tags should be journaled or if not:

* IMHO tags are the most effective way of categorizing, sorting, arrangement, grouping and isolating any kind of stuff: emails, documents, files, issues, wikis etc. The best and the simplest example of using tags is within Gmail - have you tried it? If yes maybe you will know what I am thinking about if not you can try. In just one sentence: Gmail simply replaced email folders with tags.

I think in Chiliproject we should go that way, maybe not from the beginning because it will be too difficult but for sure it should be on the target for 4.0:-)

* As I have written tagging should be treated as the only way of arrangement/isolation of all the stuff, and last but not least the issues. And to keep whole UI as simple and effective as possible no other arrangement layer should exist.

** So at the beginning they should swap out `_categories_`. Instead of categories an issue would have at least one or even a few tags - then we would solve also #465.

** Then think of using tags for trackers or rather instead of trackers. Apart from choosing `_feature_` or `_bug_` I would just add a tag `_BUG_`. We could do that also for `_priority_`. * To be compatible with current version of Chiliproject and the current idea, this tags could be obligatory ones - maybe it could be implemented as some core plugin.

** Going further, I would redesign Chiliproject so we could use tags almost like projects, that is tags could have access rights assigned to them. So adding a tag to an issue, or any other object, would cause adding access rights to that issue/object. For that we could have a special role that could be allowed to add access right to a tag. It would allow some issues to be in different "projects" at a time, which I think would be very usefull.

* Definitely there should be some mechanism for auto- or just easy tag adding. This is because some users simply do not know what tag names to add.

** So one of the examples of auto tagging is that all issues in the project should have at least one tag: their project name. That concerns also the project itself and all the subprojects.

** Another example of making life easier is autocomplete while writing a tag.

** But on the other hand there should be also a list of the tags to choose from, additionally sorted from most to least used. Or if I create object/issue from within a project or view of tags, then it inherits tags of all of his top objects.

* If I add tags for the all objects (this should definitely be the target for CP), then I can have special tab where on one page I can list all the objects with the same tags. Then changing tabs to issues, documents, files would show on these tabs only the stuff with corresponding tags (as it would be within the same project)

* I would go for quick adding tags with AJAX with submit button. Then if you followed my tagging concept, journaling could be used for tags that have their access right assigned, and maybe for obligatory tags. All other tags could stay unjournalled.

I hope I did not misuse this issue placing here the ideas I have. And I hope I was clear.

2013-11-06 09:46 pm - Matthew Connerton

Has anyone actually started working on this for core or even a CP plugin?

Files

